TRANSPARENT ENCRYPTION AND DECRYPTION WITH ALGORITHM INDEPENDENT CRYPTOGRAPHIC ENGINE THAT ALLOWS FOR CONTAINERIZATION OF ENCRYPTED FILES

Notice of Related Applications

This application is a continuation in part of U.S. Serial No. 09/074191 filed May 7, 1998, entitled "Method of Transparent Encryption and Decryption for an Electronic Document Management System," the disclosure of which is specifically incorporated herein by reference.

Field of the Invention

The present invention relates generally to cryptographic systems, and more specifically to cryptographic systems that are run by a computer program.

Background of the Invention

Global access of electronic information can be critical for even the smallest of businesses today. Very few companies operate solely within the boundaries that define their "Company." Over the last 25 years, technology has rapidly advanced and expanded these boundaries. The advent of such technologies as the Internet, Intranets, extranets, and e-mail, have made the electronic transfer of information common place in businesses today. Management of "Company" information is critical to the success of the "Company." Enterprise Document Management (EDM) and e-mail systems provide the "Company" the right technology to find any document, created in any application, by anyone, at any time, dealing with any subject, at any place in the world, and communicate to and from anyone at anytime.

With the advanced technology and integration of EDM and e-mail systems comes a wide variety of information that has varying economic values and privacy aspects. Users may not know what information is monitored or intercepted, especially when information is sent by e-mail over the Internet and outside the "Company."

E-mail is one of the fastest growing means of communication today. The use of e-mail has dramatically increased from 100,000 users in the late 1970's to about 50 million users in 1997, with over 100 million users predicted by the year 2000. This trend correlates with the advent of low-cost Internet access, mass marketed on-line services, and employer provided e-mail accounts for an estimated 30 to 40 million employees. Thus, 15% of the United States population is currently using e-mail. This number is rapidly growing. E-mail provides a quick, economical, easy to use method of sharing both thought and electronic information. Unfortunately, e-mail is like an electronic postcard for the world to see. It is transmitted across the Internet using the Simple Mail Transfer Protocol (SMTP). This protocol has virtually no security features. Messages and files can be read by anyone who comes into contact with them.

Consider the spectrum of information at risk:

- Company strategic and corporate plans (acquisitions, internal financials, sales forecasts)
- Proprietary product information (designs, formulas, processes)
- Confidential legal information (patents, client/attorney privileged information, memos)
- Private health information (test results, treatments received, lab reports)
- Private employment information (salaries, performance evaluations, benefits)

As companies increase the efficiency to access more information, their security risks will also increase. How true is this? According to a recent survey by Ernst & young LLP the following results were reported:

- 74% of the respondents say their risks have increased over the last two years.
- More than a quarter of the respondents say that their risks have increased at a faster rate than the growth of their computing.
- 73% of companies don't have the internal resources capable of dealing with network security problems.
- 55% of the respondents lacked confidence that their systems could withstand an internal attack.
- 71% of security professionals are not confident their organizations are protected from external attack.
- Two-thirds of the respondents reported losses resulting from a security breach over the last two years.

The bottom line is simple: the more information is available, the more security and authentication is needed. Increasingly, information professionals are turning to encryption and authentication technologies to ensure the privacy and integrity of "Company" information. Encryption and authentication technologies provide confidentiality, source authentication, and data integrity.

Encryption is a process of scrambling data utilizing a mathematical function called an encryption algorithm, and a key that affects the results of this mathematical function. Data, before becoming encrypted, is said to be "clear text." Encrypted data is said to be "cipher text." With most encryption algorithms, it is nearly impossible to

convert cipher text back to clear text without knowledge of the encryption key used. The strength of the encryption data is generally dependent upon the encryption algorithm and the size of the encryption key.

There are two types of encryption: symmetric (private key) and asymmetric (public key.)

Private key encryption uses a common secret key for both encryption and decryption. Private key encryption is best suited to be used in trusted work groups. It is fast and efficient, and properly secures large files. The leading private key encryption is DES (Data Encryption Standard). DES was adopted as a federal standard in 1977. It has been extensively used and is considered to be strong encryption. Other types of private key encryption include: Triple-DES, IDEA, RC4, MD5, Blowfish and Triple Blowfish.

Public key encryption uses a pair of keys, one public and one private. Each user has a personal key pair, and the user's public (or decryption) key is used by others to send encrypted messages to the user, while the private (or decryption) key is employed by the user to decrypt messages received. Public key encryption and key generation algorithms include the public domain Diffie-Hellman algorithm, the RSA algorithm invented by Riverst, Shamir and Adleman at the Massachusetts Institute of Technology (MIT), and the Pretty Good Privacy algorithm (PGP) developed by Phil Zimmermann. Because of their mathematical structure, public key encryption is slower than most private key systems, thus making them less efficient for use in a trusted network or for encrypting large files.

Although these private key and public key encryption algorithms do a good job at maintaining the confidentiality of the encrypted matter, they have numerous problems. The biggest obstacle to adoption of any type of encryption system has been ease of use. Typical encryption systems are very cumbersome. They require a user to interrupt the user's normal work flow, save the clear text document, activate the separate encryption software, and save the cipher text document under a different name. Where the subject document is ordinary e-mail contents, the process can be especially cumbersome, particularly if clear text must first be created in a separate application, then encrypted, then attached to the e-mail message.

A major concern in computing today is "total cost of ownership," or TCO. TCO recognizes that while a program might be inexpensive (or even free in the case of PGP for non-commercial use), there are significant costs in using the software. This includes the cost of installation, training, lost productivity during use and from bugs, and maintenance.

Even where one of the typical encryption systems might satisfy a user's TCO needs, it may not even be an available option. For example, typical Electronic Document Management Systems are self-contained and are not compatible with typical encryption systems.

There are many different encryption and authentication technologies that do not work with one another. This makes universal implementation of encryption systems more difficult and expensive. A need exists, therefore, for a technology that allows easy and inexpensive implementation of multiple encryption systems.

In addition, it is not always desirable to encrypt an entire document or file. For example, a memo might be sent to a group of people, but the sender might not want the entire group of people to have access to certain sensitive information contained within the memo. One way to solve this problem is to create two different memos that are sent to the two different groups. However, this practice risks inadvertent disclosure and can be cumbersome.

Another way of solving this problem is to encrypt the portion of the document that contains the sensitive information and a commercially available program allows a user to do just that. The program is told the starting and stopping point of the clear text to be encrypted, the clear text is then converted to cipher text by the encryption program, and the cipher text is then inserted back into the memo for the clear text that was encrypted. To decrypt the cipher text, a user must identify, precisely, the beginning and the end of the cipher text to be decrypted. When the cipher text has been decrypted, the program replaces the cipher text in the memo with the clear text that was originally encrypted to generate the cipher text. However, if the user makes an error in identifying the beginning or the end of the cipher text, or if the text is inadvertently modified, the decryption process will corrupt the clear text that was encrypted, thus rendering the cipher text meaningless since any subsequent attempt to decrypt the cipher text will fail.

Accordingly, there is also a need for an easy to use and inexpensive technology that allows users to conveniently encrypt and decrypt a portion of a file or document, especially if this feature can be combined with implementation of multiple encryption systems in a transparent process.

SUMMARY OF THE INVENTION

The present invention is generally directed to a method for encrypting or decrypting a file that is largely transparent to the user. This is accomplished by intercepting a change document or open document command, carrying out the encryption or decryption process, and then completing the command on an encrypted or decrypted file.

In a first, separate aspect of the present invention, one of a plurality of encryption algorithms is used to encrypt or decrypt a file. Once an encryption algorithm and an encryption key with a key value are selected, a file identifier is generated and added to the file to be encrypted. The file identifier is generated from the encryption key, an algorithm identifier associated with the selected algorithm and a data identifier associated with the file. The key value and the selected algorithm are then used to encrypt the file. The decryption process begins with the input of a decryption key with a decryption key value. The decryption key value is validated with the key value associated with the file identifier, and then the key value and the selected algorithm are used to decrypt the encrypted file.

In yet another, separate aspect of the present invention, the file to be encrypted is selected from the contents of a larger second file. The encrypted file is located in a container that can be represented in a third file that contains the portion of the second file that has not been encrypted.

Accordingly, it is a primary object of the present invention to provide a transparent cryptography process that can selectively include the features of selecting one of a plurality of encryption algorithms and allowing less than an entire file to be

encrypted and placed in a container. This and further objects and advantages will be apparent to those skilled in the art in connection with the detailed description of the preferred embodiments set forth below.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

A preferred embodiment of a method of encrypting an electronic file according to the present invention is shown in Figure 1 while a preferred embodiment of a method of decrypting an electronic file according to the present invention is shown in Figure 2. The methods can be carried out on any network capable of performing the requisite functions, as described in parent patent application Serial No. 09/074191, an individual computer, or through access to any computing device or system capable of performing the requisite functions explained below.

As used in this description, "file" is meant to include any memory resident block of computer instructions or data, including any named, structural unit of text, graphics and/or other data that can be stored, retrieved and exchanged among different computer systems and users. In this context, "memory" is meant to be defined in its broadest sense and therefore includes any storage method regardless of medium.

The encryption and decryption methods are carried out through use of a software module, referred to as a crypto server, that transparently handles the encryption of files and the decryption of encrypted files, making encryption and decryption simple and easy to use. The crypto server handles encryption and decryption without requiring user input and without normally displaying status information during normal encryption and decryption operations. The crypto server preferably includes interfaces to one or

more cryptographic systems, such as those described in the Background of the Invention section above.

Before an individual user is permitted to encrypt or decrypt a particular file in accordance with the present invention, it is desirable for the crypto server to require the user to submit to an access authentication step. Although something as simple as a user ID/password scheme can serve as an access authentication step, greater security can be provided by any number of means, or combination of means, currently known in the art or developed in the future. Examples of security devices that can be used to provide an access authentication step include a smart card or a biometric recognition system.

In an especially preferred embodiment, a user has a smart card that stores a unique user ID and password and a definable hierarchy of encryption keys. The hierarchy preferably forms a table wherein a key name is associated with each key value in the table, and the table may store both encryption keys and decryption keys as is necessary for the selected cryptographic algorithms. It should be appreciated that, in private key cryptography, the same key value is used for both encryption and decryption.

The encryption process for a particular file begins when a user issues a change document command that commands an application program to act upon the file. An example of an application program is Microsoft® Word® and examples of change document commands within that program are a "close," a "save," or a "save as" command.

Once a change document command is given, the command is translated into an "event" and the crypto server traps this event. Techniques for translating commands into events and trapping events are well known in the art and are typically different for each operating system. In Microsoft® Windows®, the event translation step comprises generating an event message.

The trapped event has the effect of alerting the crypto server that it may be necessary to encrypt the file. However, preferably before encrypting the file, the crypto server tests whether the file should be encrypted. The crypto server may also invoke an option to initiate a virus scan program or initiate a virus scan program to run a virus scan on the file before it is encrypted.

One test that the crypto server may run to determine whether a file should be encrypted is to determine whether the user has been authenticated. If a smart card or similar means is used for storing keys, this test is necessary because the keys will not even be available unless the user was authenticated. Another test that may be run is to determine whether the file was already encrypted when it was opened within the application program. By default, a file that was already encrypted when opened should be encrypted when closed or saved. Another test that may be run is to check a database to determine if the file meets a predetermined criteria for invoking encryption, an example of which is explained in greater detail in connection with Electronic Document Management systems in parent application Serial No. 09/074191.

If for any reason the file is not to be encrypted, then the crypto server passes control of the file back to the application program which then performs the change document command on the file. Alternatively, the decision not to encrypt, for one or

more reasons, may result in an error message being displayed to the user, and may result in the file not being closed or saved. At this point, for files that are not to be encrypted, the encryption method is complete.

If the file is to be encrypted, then the crypto server preferably obtains an encryption key name that is associated with the file.

The crypto server then uses the encryption key name to retrieve an encryption key value that is associated with the key name. For most encryption algorithms, the encryption key is a multi-digit number that is difficult to remember and even difficult to transcribe. The encryption key name is preferably an alphanumeric descriptor that may be used by the user or a system administrator for administering the encryption key value. Preferably, the encryption key value is also related to the identity of the user, and this can be accomplished by retrieving the encryption key value from a key table stored in the user's smart card or a secure file that is associated with the relevant encryption key name.

Once the crypto server has the encryption key value, the crypto server then encrypts the file with encryption key value, and passes control of the file back to the application program so that the change document command can be executed. At this point, for files that are to be encrypted, the encryption method is complete.

The decryption process for a particular file begins when a user issues an open document command that commands an application program to act upon the file. An example of an application program is Microsoft® Word® and an example of an open document command within that program is an "open" command.

Once an open document command is given, the command is translated into an "event" and the crypto server traps this event. The trapped event has the effect of alerting the crypto server that it may be necessary to decrypt the file. However, preferably before decrypting the file, the crypto server tests whether the file should be decrypted. Preferably, these tests are complimentary to those described above with respect to the encryption process. The crypto server may also invoke an option to initiate a virus scan program or initiate a virus scan program to run a virus scan on the file after it is encrypted.

If for any reason the file is not to be decrypted, then the crypto server passes control of the file back to the application program which then performs the open document command on the file. Alternatively, the decision not to decrypt, for one or more reasons, may result in an error message being displayed to the user, and may result in the file not being opened. At this point, for files that are not to be decrypted, the decryption method is complete.

If the file is to be decrypted, then the crypto server preferably obtains a decryption key name that is associated with the file. The decryption key name is preferably obtained from the file's header or from an encrypted files table.

The crypto server then uses the decryption key name to retrieve a decryption key value that is associated with the decryption key name. Preferably, the decryption key value, like the encryption key value, is also related to the identity of the user, and this can be accomplished by retrieving the decryption key value from the key table stored in the user's smart card or a secure file associated with the decryption key name.

Once the crypto server has the decryption key value, the crypto server then decrypts the file with the decryption key value. The crypto server may also invoke an option to initiate a virus scan program or initiate a virus scan program to run a virus scan on an encrypted or on a decrypted file. After the crypto server has completed decryption of the encrypted file it passes control of the file back to the application program so that the open document command can be executed. At this point, for files that are to be decrypted, the decryption method is complete.

The foregoing description sets forth a preferred embodiment of a cryptographic process that is largely transparent to a user which is accomplished by intercepting a change document or open document command, carrying out an encryption or decryption process, and then completing the command on an encrypted or decrypted file. In an especially preferred embodiment, this cryptographic processes is modified so that the crypto module is able to select from a plurality of encryption algorithms, and this particular feature can be used in other cryptographic processes as well. This particular feature will now be described in greater detail.

The crypto module can be programmed to select one of a plurality of encryption algorithms according to a preselected criteria or a preselected algorithm. An example of a simple, preselected criteria is to encrypt all files of a certain type, or all files encrypted within a certain time frame, with a chosen algorithm. An example of a simple, preselected algorithm is to chose the preselected algorithm from a set of algorithms by simple rotation. For example, if there are three algorithms in the set, the crypto module could encrypt a first file with the first algorithm, a second file with the second algorithm,

a third file with the third algorithm, a fourth file with the first algorithm, and so forth, for a preselected amount of time or through a preselected number of rotations.

Once the encryption algorithm that will be used with a file is selected, the crypto module generates a file identifier from the encryption key, an algorithm identifier associated with the algorithm, and a data identifier associated with the file. The file identifier is then inserted into the file by the crypto module according to a preselected criteria or a preselected algorithm. The details of such insertion can serve to create additional security, and such details would be known by a person of ordinary skill in the art of computer programming.

During the decryption process, the crypto module obtains the encryption key and the algorithm identifier from the file identifier. The encryption key is compared to the decryption key that is input into the crypto module and the decryption key is validated if it is the same as the encryption key. If the decryption key is validated, the crypto module decrypts the encrypted file by using the validated decryption key and the algorithm identified by the algorithm identifier.

The integrity of the foregoing cryptography process can be validated by uniquely identifying the encrypted file with an encrypted data identifier during encryption and testing the encrypted data identifier after decryption by regenerating the encrypted data identifier and ascertaining that they are the same.

Additional security for the foregoing cryptography process can be provided by separately encrypting either a portion of the file identifier or the entire file identifier before it is inserted into the file to be encryped, and then decrypting whatever portion of the file identifier has been encrypted during the decryption process.

In another especially preferred embodiment, the cryptographic process allows just a portion of a file to be encrypted and placed in a "container." In the context of this invention, a container is any way in which data or program code can be represented in a file when it is not part of the file. As part of the encryption process, a file is selected from within the contents of a second file that contains more information than the file. The contents of the file is then placed in a container and a third file is created that contains the container and that portion of the second file that is not included in the file. The container can be represented within the third file by an object linking and embedding ("OLE") container object or other representation supported by the file. During the decryption process, the encrypted file is removed from the container, decrypted and then preferably reinserted into the third file to recreate the second file.

The above discussion of this invention is directed primarily to the preferred embodiments and practices thereof. Further modifications are also possible without departing from the inventive concepts described herein. For example, files to be encrypted, or encrypted files, can be located in indexed document or image repositories. In addition, the invention is particularly well suited to the application of sending the encrypted file from a first person to a second person (even if the second person is the same as the first person) by electronic messaging, such as e-mail, over the Internet or any other data transfer over a network.

Accordingly, it will be readily apparent to those skilled in the art that still further changes and modifications in the actual implementation of the concepts described herein can readily be made without departing from the spirit and scope of the invention as defined by the lawful scope of the following claims.